# Endings Principles for Digital Longevity

We divide digital projects into five primary components: data, products, processing, documentation, and release management.

## 1. DATA

Data is the expression of the source information, knowledge, and expertise of our researchers. The following principles apply to data:

**1.1** Data is stored only in formats which conform to open standards and which are amenable to processing (TEI XML, GML, ODF, TXT).

**1.2** Data is subject to version control (Subversion, Git).

**1.3** Data is continually subject to validation and diagnostic analysis.

## 2. PRODUCTS

Products are the project output intended for end-users, typically in the form of websites or print documents. The following principles apply to products intended for the web:

**2.1** No dependence on server-side software: build a static website with no databases, no PHP, no Python.

**2.2** No boutique or fashionable technologies: use only standards with support across all platforms, whose long-term viability is assured. Our choices are HTML5, JavaScript and CSS.

**2.3** No dependence on external libraries: no JQuery, no AngularJS, no Bootstrap.

**2.4** No query strings: every entity in the site has a unique page with a simple URL.

**2.5** Graceful failure: every page should still basically work even in the absence of JavaScript or CSS support.

**2.6** Massive redundancy: every page contains all the components it needs, so that it will function without the rest of the site if necessary, even though this means duplicating information across the site.

**2.7** Relentless validation: every site build involves validation of all input data (XML) and all output code (HTML5, JavaScript, CSS).

**2.8** Inclusion of data: every site should include a documented copy of the source data, so that users of the site can repurpose the work easily.

These principles are tempered by the following concessions:

**2.9** Once a fully-working static site is achieved, it may be enhanced by the use of other services such as a server-side indexing tool (Solr, eXist) to support searching and similar functionality.

**2.10** The use of an external library may be necessary to support a specific function which is too complex to be coded locally (such as mapping or cryptography). Any such libraries must be open-source and widely-used, and must not themselves have dependencies.

## 3. PROCESSING

Processing code is written and maintained by the project technical staff, and is also subject to version control. Processing code provides all the following functions:

**3.1** Validation of data.

**3.2** Diagnostics (analysis of data to identify issues, and to track and evaluate progress).

**3.3** Generation of products.

**3.4** Validation of products.

For larger projects, these processes should all be triggered automatically by any change to version-controlled resources, typically running on a continuous integration server. Processing code should be open-source where possible, and have minimal dependencies, but it is not expected to have significant longevity; data and products are designed to survive, but processing code is contingent.

## 4. DOCUMENTATION

**4.1** Data models, including field names, descriptions, and controlled values, should be clearly documented in a static document that is maintained with the data, and should also form part of the products.

**4.2** All rights and intellectual property issues should be clearly documented. Where possible the Data and Products should be released under open licenses (Creative Commons, GNU, BSD, MPL).

## 5. RELEASE MANAGEMENT

Release management handles the public release of products. Without good release management, a project can never end gracefully; it can only falter and die. These principles apply to release management:

**5.1** Releases should be periodical and carefully planned. The "rolling release" model should be avoided.

**5.2** A release should only be made when the entire product set is coherent, consistent and complete (passing all validation and diagnostic tests).

**5.3** Like editions of print works, each release of a web resource should be clearly identified on every page by its build date and some kind of version number.

**5.4** Web resources should include detailed instructions for citation, so that end-users can unambiguously cite a specific page from a specific edition.